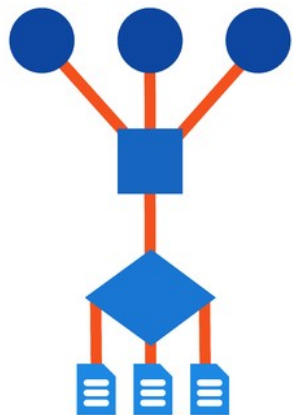


JavaScript

Prof. Daniel Di Domenico



Sintaxe dos comandos

JavaScript

- A linguagem JavaScript possui comandos para controlar o fluxo de execução em código, sendo:
 - Comando **IF** (se) para condições
 - Comando **ELSE** (senão) para condições compostas
 - Comando **SWITCH** (escolha) para condições com casos fixos com uma única entrada
 - Comando **WHILE** (enquanto) para repetições controladas por condições
 - Comando **FOR** (para) para repetições considerando um intervalo numérico pré-definido

IF e ELSE (se e senão)

- Se a **condição** for verdadeira, o bloco é executado
 - ELSE é o caso contrário, sendo executado se as demais condições testadas nos IFs forem falsas

//IF simples

var n1 = 1;

var n2 = 2;

if(n1 == 1 && n2 >= 2) {
 alert("n1 = 1 e n2 >= 2");
}

Condição

Em JS, as **chaves** para abrir e fechar o bloco são opcionais caso exista apenas uma linha a ser executada

//IF composto

var n1 = 1;

if(n1 == 1) {
 alert("n1 = 1");
} **else if**(n1 == 2) {
 alert("n1 = 2");
} **else** {
 alert("n1 != 1 e n1 != 2");
}

Operadores

- Operadores relacionais:
 - == (igual)
 - === (idêntico – igual e de mesmo tipo)
 - !=, <> (diferente)
 - !== (não idêntico)
 - > (maior)
 - < (menor)
 - >= (maior ou igual)
 - <= (menor ou igual)
- Operadores lógicos
 - && (E)
 - || (OU)
 - ! (NEGAÇÃO - not)

SWITCH (escolha)

- Se a **variável** for igual ao **caso**, são executados os comandos do **bloco**
 - DEFAULT é o caso contrário. O uso do BREAK é obrigatório para finalizar o fluxo do comando SWITCH

Variável de
teste para os
casos

```
var n1 = 1;

switch(n1) {
  case 1:
    alert("n1 = 1");
    break;
  case 2:
    alert("n1 = 2");
    break;
  default:
    alert("n1 != 1 e n1 != 2");
}
```

WHILE (enquanto)

- **Repete** a execução de um bloco de comandos enquanto a condição for **verdadeira**

```
var n1 = 1;
```

//while com condição avaliada no INÍCIO

```
while(n1 <= 5) {
```

```
  console.log(n1);
```

```
  n1 = n1 + 1;
```

```
}
```

Condição

Executa **ao menos uma vez**, mesmo a condição sendo inicialmente FALSA

```
var n1 = 1;
```

//while com condição avaliada no FINAL

```
do {
```

```
  console.log(n1);
```

```
  n1 = n1 + 1;
```

```
} while(n1 <= 5);
```

Condição

Saída na tela (ambos):

1
2
3
4
5

FOR (para)

- **Repete** a execução de um bloco de comandos enquanto a condição for **verdadeira**
 - Após cada iteração, ocorre a execução de uma expressão de incremento

2- Condição de parada da repetição

$n1++$ equivale a
 $n1 = n1 + 1$

```
for( var n1=1; n1<=5; n1++ ) {  
    console.log(n1);  
}
```

1- Declaração da variável de controle da repetição

Saída na tela:

1
2
3
4
5

3- Incremento da variável de controle